UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/065,801	11/20/2002	Cynthia Bertini	201-1116 (81045450)	6565

28549 7590 01/12/2007

ARTZ & ARTZ, P.C.
28333 TELEGRAPH ROAD, SUITE 250
SOUTHFIELD, MI 48034

EXAMINER	
PATEL, SHAMBHAVI K	
ART UNIT	PAPER NUMBER
2128	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	01/12/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/065,801
Filing Date: November 20, 2002
Appellant(s): BERTINI ET AL.

John A. Artz
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 16 October 2006 appealing from the Office action mailed 05 May 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,289,319	**Heile**	8-1999
6,094,654	**Van Huben**	6-1998

Tou et al. "Knowledge-Based Approach for the Verification of CAD Database Generated by an Automatic Schematic Capture System" 24th ACM/IEEE Design Automation Conference: 1987.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. **Claims 1, 3-6, 8-12, 14-19, and 21-24 are rejected under 35 U.S.C. 102(e)** as being anticipated by Heile et al, herein referred to as '**Heile' (US Patent No. 6,289,319)**.

Regarding claim 1:

Heile discloses a method for managing electrical schematic data comprising:

a. creating a logical schematic for a part **(column 6 lines 54-55, column 7 lines 14-22).**

b. creating a layout schematic for said part **(column 7 lines 49-52).** Heile teaches that before developing the design, a system specification for the design is obtained. The specification includes information on the pins, system functionality, timing restraints, etc. **(column 6 lines 8-13)**. Based on this, the design templates are generated **(column 6 lines 28-30)**, and then the blocks are implemented in a top-down method **(column 6 lines 55-56).** After each of the blocks has been implemented, they are compiled. Part of the compilation includes doing a schematic vs. layout check **(column 7 lines 49-53).** In order to do this, a layout has to be generated for each of the blocks.

c. creating a physical schematic for said part **(column 7 lines 52-57).**

d. associating said logical schematic, said layout schematic, and said physical schematic together to from a part master file **(column 5 lines 41-47).** The project taught by Heile contains all project files, design files, assignment files, and simulation files, among other

things. *Thus, a project is functionally equivalent to the part master file claimed by the applicant.*

e. storing said part master file on a computer network **(column 8 lines 35-49)**

f. providing access to said part master file to a plurality of user locations **(column 8 lines 35-49)**

g. controlling modifications of said part master file whereby said controlling comprises allowing only one of said plurality of user locations to modify said part master file at a time **(column 3 lines 19-21).**

Regarding claim 3:

Heile discloses the method of claim 1, further comprising the steps of modifying said part master file and tracking a modification to said part master file **(column 10 lines 20-25, 28-32, column 11 lines 1-12)**. The method taught by Heile allows for users to retrieve older versions of a file **(column 13 lines 51-53)**. In order to do this, the changes that are made to a file must be tracked.

Regarding claim 4:

Heile discloses the method of claim 3, wherein said tracking comprises storing a revised part master file **(column 5 lines 46-47, column 10 lines 28-32, column 13 lines 32-34)**. The project taught by Heile also holds a project database, which is responsible for storing intermediate data structures and version information **(column 5 lines 46-47)**. In order to rollback to older versions of a file **(column 13 lines 51-53)**, the original version of the file must be saved in the project, along with the new revised version.

Regarding claim 5:

Heile discloses the method of claim 3, further comprising the step of notifying an interested user location of said modifying **(column 3 lines 22-26).**

Regarding claim 6:

Heile discloses the method of claim 1, wherein said plurality of user locations comprises at least one remote user location **(column 8 lines 42-49, 53-57).**

Regarding claim 8:

Heile discloses the method of claim 1, wherein said associating is accomplished by a computer software program **(column 5 lines 40-47, column 6 lines 13-15).** The entire design is electronic, and all files contained within a project are associated to be part of the same design.

Regarding claim 9:

Heile discloses the method of claim 1, further comprising the steps of creating a second logical schematic for a sub-part, creating a second physical schematic for said sub-part, creating a second layout schematic for said sub-part, associating said second logical schematic, said second physical schematic and said second layout schematic together to form a sub-part master file, and storing said sub-part master file in said part master file **(column 6 lines 54-61, column 7 lines 6-13, 39-42).** Heile recognizes that for complicated designs, there are multi-level block diagrams (blocks within blocks). After the top-level diagram is done, the other blocks are then implemented in an order specified by the engineer (column 7 lines 6-10). Each of the blocks is implemented using the same methodology as the top-level block **(column 7 lines 11-13).**

Regarding claim 10:

Heile discloses the method of claim 9, further comprising the step of controlling modification of said sub-part master file, allowing one of said plurality of user locations to modify said sub-part master file at a time **(column 3 lines 19-21)**. The examiner interprets a sub-part master file to indicate any file that is contained within the master file. This is functionally equivalent to the global source file taught by Heile.

Regarding claim 11:

Heile discloses a system for managing electrical schematic data comprising:

a. a computer **(column 25 lines 65-67, column 26 lines 1-2)**.

b. at least one computer aided engineering software program **(column 1 lines 59-62, column 2 lines 61-66)**, said at least one software program being capable of creating a logical schematic **(column 6 lines 54-55, column 7 lines 14-22)**, a layout schematic **(column 7 lines 49-52)**, and a physical schematic **(column 7 lines 52-57, 34-35)** for a part based on an input into said computer from a user **(column 6 lines 8-15)**. Heile teaches that before developing the design, a system specification for the design is obtained. The specification includes information on the pins, system functionality, timing restraints, etc. **(column 6 lines 8-13)**. Based on this, the design templates are generated **(column 6 lines 28-30)**, and then the blocks are implemented in a top-down method **(column 6 lines 55-56)**. After each of the blocks has been implemented, they are compiled. Part of the compilation includes doing a schematic vs. layout check **(column 7 lines 49-53)**. In order to do this, a layout has to be generated for each of the blocks.

c. a computer schematic management utility, said computer schematic management utility being capable of associating said logical schematic, said layout schematic, and said physical schematic together to form a part master file **(column 5 lines 40-47, column 6 lines 13-15)**. The project taught by Heile contains all project files, design files, assignment files, and

simulation files, among other things. *Thus, a project is functionally equivalent to the part master file claimed by the applicant.* The entire design is electronic, and all files contained within a project are associated to be part of the same design.

d. a computer network, said computer network comprising said computer and a plurality of user locations, aid computer network being capable of storing said part master file and providing access to said part master file to said plurality of user locations **(column 8 lines 35-49)**

e. whereby said computer schematic management utility controls modification of said part master file stored on said computer network **(column 3 lines 19-26, column 10 lines 20-25, 28-35)**. The utility controls modifications by making sure only one user edits a global file at a time, tracking the changes, and notifying users of such changes.

Regarding claim 12:

Heile discloses the system of claim 11, wherein said computer schematic management utility controls modification of said part master file by allowing only one of said plurality of user locations to modify said part master file at a time **(column 3 lines 19-21).**

Regarding claim 14:

Heile discloses the system of claim 11, wherein said computer schematic management utility is further capable of tracking a modification of said part master file **(column 10 lines 20-25, 28-32)**. The system taught by Heile allows users to retrieve older versions of a file **(column 13 lines 51-53)**. In order to do this, the changes that are made to a file must be tracked.

Regarding claim 15:

Heile discloses the system of claim 14, wherein said tracking comprises storing a revised part master file **(column 5 lines 46-47, column 10 lines 28-32, column 13 lines 32-34)**. The project taught by Heile also holds a project database, which is responsible for storing intermediate data structures and version information **(column 5 lines 46-47).** In order to rollback to older versions of a file **(column 13 lines 51-53),** the original version of the file must be saved in the project, along with the new revised version.

Regarding claim 16:

Heile discloses the system of claim 11, further comprising an interested user list, wherein said computer schematic management utility generates a notification to interested user list when said part master file is modified **(column 14 lines 23-34)**. By placing the work space in default mode, the system will automatically retrieve updated files for the user.

Regarding claim 17:

Heile discloses the system of claim 16, wherein said notification comprises an electronic message **(column 14 lines 23-34)**. By placing the work space in default mode, the system will automatically retrieve updated files for the user.

Regarding claim 18:

Heile discloses the system of claim 11, wherein said plurality of user locations comprises at least one remote user location **(column 8 lines 42-49, 53-57).**

Regarding claim 19:

Heile discloses a method for managing electrical schematic data comprising:

a. creating a logical schematic for a part with a first computer aided design tool **(column 6 lines 54-55, column 7 lines 14-22)**

b. creating a layout schematic for said part based on said logical schematic with a second computer aided design tool **(column 7 lines 49-52).** Heile teaches that before developing the design, a system specification for the design is obtained. The specification includes information on the pins, system functionality, timing restraints, etc. **(column 6 lines 8-13).** Based on this, the design templates are generated **(column 6 lines 28-30)**, and then the blocks are implemented in a top-down method **(column 6 lines 55-56).** After each of the blocks has been implemented, they are compiled. Part of the compilation includes doing a schematic vs. layout check **(column 7 lines 49-53).** In order to do this, a layout has to be generated for each of the blocks.

c. creating a physical schematic for said part based on said logical schematic and said layout schematic with a third computer aided design tool **(column 7 lines 52-57, 34-35).**

d. associating said logical schematic, said layout schematic, and said physical schematic together to form a part master file, said associating comprising storing said logical schematic, said layout schematic, and said physical schematic in a single file **(column 5 lines 40-47, column 6 lines 13-15).** The project taught by Heile contains all project files, design files, assignment files, and simulation files, among other things. Thus, a project is functionally equivalent to the part master file claimed by the applicant. The entire design is electronic, and all files contained within a project are associated to be part of the same design.

e. storing said part master file on a computer network **(column 8 lines 35-49)**

f. providing access to said part master file to a plurality of user locations **(column 8 lines 35-49)**

g. controlling modification of said part master file, whereby said controlling comprises allowing only one of said plurality of user locations to modify said part master file at a time **(column 3 lines 19-21).**

Regarding claim 21:

Heile discloses the method of claim 19, further comprising the steps of modifying said part master file and tracking a modification to said part file **(column 10 lines 20-25, 28-32).** The method taught by Heile allows for users to retrieve older versions of a file **(column 13 lines 51-53).** In order to do this, the changes that are made to a file must be tracked.

Regarding claim 22:

Heile discloses the method of claim 21, wherein said tracking comprises storing a revised part master file **(column 5 lines 46-47, column 10 lines 28-32, column 13 lines 32-34).** The project taught by Heile also holds a project database, which is responsible for storing intermediate data structures and version information **(column 5 lines 46-47).** In order to rollback to older versions of a file **(column 13 lines 51-53)**, the original version of the file must be saved in the project, along with the new revised version..

Regarding claim 23:

Heile discloses the method of claim 21, further comprising the step of notifying an interested user location of said modifying **(column 3 lines 22-26, column 14 lines 23-27).**

Regarding claim 24:

Heile discloses the method of claim 19, wherein said plurality of user locations comprises at least one remote user location **(column 8 lines 42-49, lines 53-57).**

2. **Claim 7 is rejected under 35 U.S.C. 103(a)** as being unpatentable over **Heile in view of Van Huben et al (US Patent No. 6,094,654)**, herein referred to as "Van Huben".

Regarding claim 7:

Heile fails to explicitly disclose the method of claim 1, further comprising the step of storing a pointer in said part master file, said pointer being capable of indicating a storage location of said logical schematic, said physical schematic, and said layout schematic.

Van Huben teaches a similar method where in the case of the design data, the physical data is tracked via pointers whenever possible **(Van Huben column 13 lines 20-22)**. The design data disclosed by Van Huben is analogous to the design files contained in the project taught by Heile.

At the time of the invention, it would have been obvious to one skilled in the art to combine the teachings of Heile and Van Huben to minimize the amount of file movement between servers and to minimize the bottleneck caused by too many users accessing the servers **(Van Huben column 13 lines 20-22).**

3. **Claims 2, 13 and 20 are rejected under 35 U.S.C. 103(a)** as being unpatentable over **Heile in view of Tou et al (Knowledge-Based Approach for the Verification of CAD Database Generated by an Automatic Schematic Capture System)**, herein referred to as Tou.

Regarding claims 2, 13 and 20:

Heile discloses using a project to hold all files that are part of a design **(column 5 lines 41-47). Heile fails to explicitly disclose** using the logical schematic, physical schematic, and layout schematic to form a schematic image file.

Tou teaches using the schematics of a design to form schematic image files **(Tou page 1 section I lines 6-7, 10-12, page 2 section II lines 1-4).**

At the time of the invention it would have been obvious to one skilled in the art to combine the teachings of Tou and Heile to provide a portable version of the design. The schematics produced by the engineering design aid software are not viewable across all platforms, software products, etc. By producing an image file of the schematics, users across all platforms and users who do not have the engineering design aid software can view the design. Furthermore, automating the schematic image capturing process improves the efficiency and productivity of Design Automation **(Tou: section I 'Introduction' paragraph 2).**

(10) Response to Argument

Response to Argument – Prior Art Rejections

Appellant's arguments pertaining to the 102 and 103 rejections are not persuasive. Appellant's arguments focusing on claims 1-24 are addressed in the order in which they are presented on the Appeal Brief starting on page 4.

The main thrust of Appellant's arguments center around Heile's alleged failure to disclose a layout schematic and a physical schematic. The Examiner emphasizes that the secondary reference used for the rejection of claim 7 is relied upon only for the use of a pointer that indicates the storage location of data, and the secondary reference used for the rejection of claims 2, 13 and 20 is relied upon only for the teaching of image files. The Examiner maintains that the above features are taught by Heile, Van Huben and Tou, as shown below.

(10.1) Appellant argues, on page 4, that Heile does not teach a layout schematic and a physical schematic.

Examiner's Answer:

The Examiner notes that the term "layout schematic" is defined in paragraph [0017] of the specification as a schematic diagram that shows the electrical components used in the design and the connections between them. Appellant is directed to **figure 17** of the Heile reference, which shows electrical components and their interconnections. Appellant is further directed to **column 7 lines 52-57, 34-35** of the Heile reference, which recites a top-level block diagram that shows complete design connectivity and completing a layout vs. schematic check.

The Examiner notes that the term "physical schematic" is defined as a schematic diagram that fully describes the electrical components used and their interconnections, showing the exact physical position of the components. Appellant is directed to **figure 18** of the Heile reference, which shows the shape of the overall design. Appellant is further directed to **column 7 lines 48-57** of the Heile reference, which states performing a *place and route*, (a stage in circuit design at which components are graphically placed on the board and the wires drawn between them).

(10.2) Appellant argues, on page 5, that Van Huben does not teach or suggest the step of storing a pointer in a part master file wherein the pointer indicates a storage location of a logical schematic, a physical schematic, and a layout schematic.

Examiner's Answer:

The Examiner notes that the Van Huben reference is relied upon only for the use of a pointer that indicates the storage location of data. As per the Appellant's admission (page 5), Van Huben discloses a pointer that indicates a storage location of data. The Examiner notes that the use of pointers in the storage

of data is not exclusive to any one specific technology or application; it is widely used in all fields and technologies.

(10.3) Appellant argues, on page 6, that Tou does not teach the creation of a part master file from a logical schematic, layout schematic, and physical schematic.

Examiner's Answer:

The Examiner notes that Tou is not relied upon for the teaching of a master file. Heile (the primary reference) discloses the creation of a master file from a logical, layout and physical schematic (see rejection of claim 1 above). Tou is relied on solely for his teaching of creating image files based on schematics in the circuit design process **(see section II lines 1-3 of Tou)**. The Examiner notes that the portion of Tou cited in the rejection recites the formation of a schematic image file (while Tou does teach converting the image file into a data file, this step in not pertinent to the rejection).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Shambhavi Patel, Patent Examiner

Conferees:

Kamini Shah, Supervisory Patent Examiner

Eddie Lee, Supervisory Patent Examiner

KNOWLEDGE-BASED APPROACH FOR THE VERIFICATION OF CAD DATABASE GENERATED BY AN AUTOMATIC SCHEMATIC CAPTURE SYSTEM

J.T. Tou W.H. Li K.C. Fan C.L. Huang

Center for Information Research
University of Florida
Gainesville, Florida

Abstract

CAD database generated by an automatic schematic capture system needs to be verified before it can be used in design automation. This verification is best performed by a knowledge-based expert system. Presented in this paper is the design of a knowledge-based system for the verification of CAD database generated by AUTORED. Database-driven, pattern-directed inference technique is employed to identify and correct erroneous data records due to misrecognition. This knowledge-based verification system has been implemented on a VAX 11/750 equipped with a video camera to read circuit diagrams into the computer. Integration of this verification system into AUTORED makes the automatically generated database accurate for design automation applications.

I. INTRODUCTION

In recent years, CAD/Design Automation techniques have become standard tools for the design and testing of complex electronic circuits and systems. Numerous CAD software packages are available commercially. One of the remaining major problems to be solved is automatic capture of schematics by computer. As we know, creative designers often initiate the design by expressing their new ideas in schematic diagrams. However, most CAD software developed today cannot read schematic diagrams; it can only handle alphanumeric symbols. In current commercially available CAD systems, capture of electronic circuit diagrams is performed by an interactive editor through a special device such as light pen, tablet [1,2]. To automatically convert a schematic diagram into CAD database is a necessary operation for improving the efficiency and productivity of Design Automation.

We all realize that data entry by human operators is time-consuming and error-prone. Automated machines for reading and interpreting schematic diagrams are needed for enhancing the current CAD capabilities. In response to this need, the Center for Information Research has developed the AUTORED system to read creative circuit design into the computer for CAD work [3,4]. AUTORED is the acronym for Automatic Reading of Diagrams, which is an automatic schematics capture system. Making use of image understanding techniques, AUTORED is designed to read and interpret electronic circuit diagrams from their images. Figure 1 describes the software architecture of the AUTORED system. The accuracy and completeness of the interpretation results depend upon the quality of the raw images of the circuits diagram. To guarantee high reliability and accuracy, automatic verification of the captured database is needed.

This paper presents an approach to automatic verification of CAD database which is generated by the schematic capture system. Our approach makes use of knowledge-based expert systems [5-8] to detect and correct interpretation errors which are caused by the fuzziness of schematic images or errors in the circuit diagram. Knowledge about electronic and logic circuit theory is encoded as inference rules to correct image-level recognition and interpretation errors in the same manner that an electronic engineer would do.

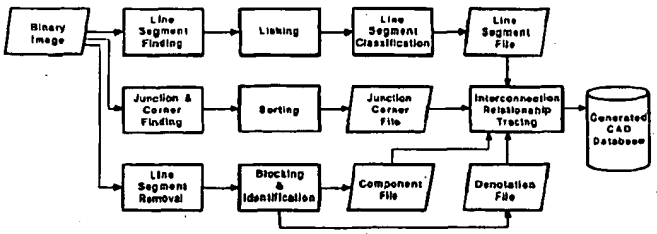


Figure 1. Software Architecture of the AUTORED System

II. ORGANIZATION OF CAD DATABASE GENERATED BY AUTORED

The captured schematic data are organized in CAD database as shown in Figure 2. The organization consists of four data files which represent the geometric, electrical, and interconnection relationships of the circuit diagram. These data files are

(1) Component data file
(2) Pin connection data file
(3) Line data file
(4) Junction data file

The component data file contains the following types of data:

* Database access information – component ID, name
* Geometric properties -- coordinates, orientation, pin data pointer
* Electrical properties -- component category, type, number of pins, denotation/value

The pin connection data file represents interconnection relationships between component pins and connection line segments. The line data file contains interconnection relationships between connection line segments and junctions or component pins. The junction data file contains interconnection relationships between junctions and connection line segments.

III. ERRORS IN CAPTURED SCHEMATIC DATA

Following the procedures outlined in Figure 1, the AUTORED system processes the schematic images, interprets the image data, and converts the captured schematics into machine readable CAD data files. When the schematic images are crisp and free from errors, the AUTORED system is able to generate CAD data files accurately. However, most schematic images are fuzzy and may contain some errors. Consequently, the generated CAD database is by no means error free. It is necessary to verify the CAD database, before the captured schematic data will be used in the design.

Poor quality of images may cause three types of errors in the CAD database:

(1) Incomplete data records. If some part of the schematic image is smeared or missing after preprocessing, the circuit elements become unrecognizable or they can only be partially recognized. This situation will cause incomplete or empty data records in the generated CAD database.

(2) Redundant data records. If noise in the schematic image is not completely removed, it may be recognized as simple circuit elements such as junction points, arrowheads, line segments, etc. This situation will result in redundant or superfluous data records in the generated CAD database.

(3) Erroneous data records. If the input schematic is incorrect or the schematic image is somehow distorted, the circuit elements will be misrecognized. This situation will cause erroneous data entries in the generated CAD database.

Type 1 error may be detected by checking the missing fields of the data record. Type 2 error may be detected by finding all inconsistencies. Type 3 error may be detected by examining the interconnection relations. In our approach the system is designed to perform the following database verifications:

(1) Data completeness verification
(2) Data consistency verification
(3) Interconnection verification based on circuit theory

In data completeness verification, all the fields in a record are checked to determine whether they are filled with data. An empty data entry signifies an error. Data consistency verification is mainly used for checking errors in the component data file. The geometric and electric properties of each generated component data record are compared with the standard descriptions stored in the component library. Any inconsistency reveals some error in the data entry. Interconnection verification examines the interconnection relations of the captured circuit to determine any violation of circuit theory. Since all circuit elements are directly or indirectly connected, any misinterpretation of a circuit element will propagate throughout the circuit network. Erroneous interconnections may result in discrepancies from theoretical values. For example, if a resistor is misrecognized as an inductor, all the connected circuit elements will show wrong interconnection information and violation of circuit theory is revealed.

Detecting Errors in Line Data File

Automated recognition of connection lines in a circuit diagram may result in three types of errors: (1) broken lines, (2) missing lines, (3) redundant lines. These errors will appear in the database as a blank, redundant, or erroneous entries. Figure 3 illustrates several cases of recognition errors of connection lines. In case 1, a broken line is recognized as two lines. Each line now has only one connection. Both component A and B have one terminal connection missing. In case 2 and 3, a missing line also causes the components to have missing terminal connections. In case 4, an isolated line creates a line data record which has no connections. In case 5 and 6, noise is misrecognized as an isolated group of redundant lines and components. Components

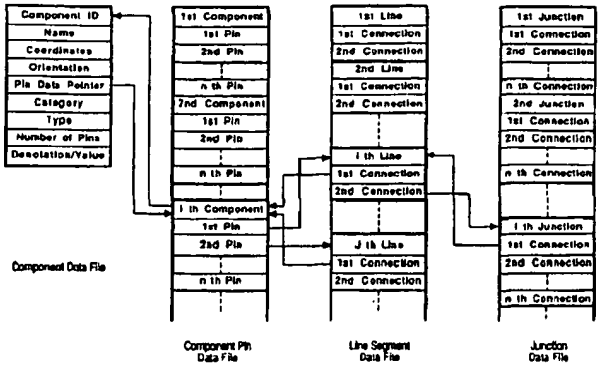


Figure 2. Generated CAD Database Organization

A and B, which are connected to line 1, have only internal connections within the group. These errors can be detected by examining the data entries of the related circuit elements, via data completeness verification.

Detecting Errors in Junction Data File

Recognition errors in circuit junction interpretation are similar to the errors in connection lines interpretation. The line crossing patterns may be mis-recognized as a missing junction or a superfluous junction as shown in Figure 4. In case (1), a missing junction creates four broken lines, resulting in missing terminal connection for each of the four components A,B,C, and D. In case (2), a missing junction changes the interconnection of the four components. In case (3), a superfluous junction distorts the circuit connections.

The corner connection and the T junction patterns may be mis-recognized as a missing junction resulting in disconnection of line segments, as shown in case (4) - (6). In case (7), a noise image or denotation is mis-recognized as a junction and a line segment which form an isolated subgraph. The recognition errors may be detected by data completeness verification, except case (2), (3), and (5). In these three cases, the circuit appears to have the right number of connections, but the interconnection relationships are mis-interpreted. Interconnection verification technique will be used to detect these recognition errors. To illustrate this idea we consider the circuit of Figure 5 which shows an example of the interconnection process. Assume that the AUTORED system mis-recognized a junction at location A. The generated CAD database will indicate that both terminals of the resistor R are connected to the same reference voltage. This is contradictory to circuit theory. This superfluous junction is deleted.

Detecting Data in Component Data File

A circuit component may be partially recognized or mis-recognized. The former will cause incomplete component data records, and the latter will create erroneous component data records. Incomplete data records can be detected by data completeness verification. Erroneous data records will be detected by data consistency verification or by circuit interconnection verification. Figure 6 illustrates an example of detecting an erroneous data record by interconnection verification. In this circuit a Zener diode is mis-recognized as an ordinary diode which creates an illegal interconnection contradictory to circuit theory.

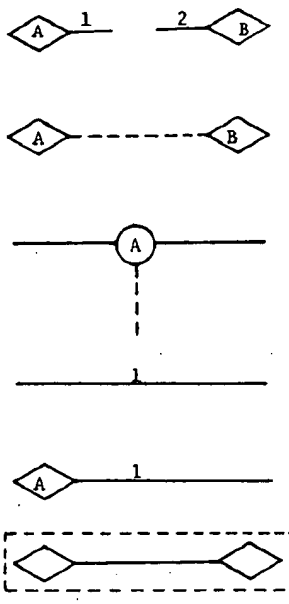
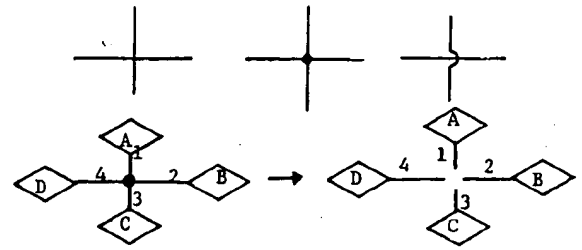
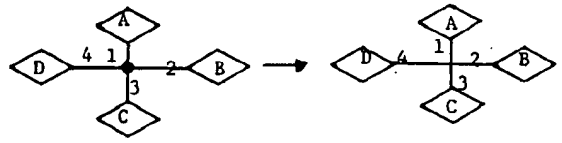


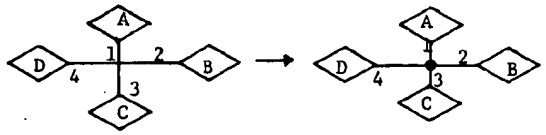
Figure 3. Possible errors in interpreting connection lines



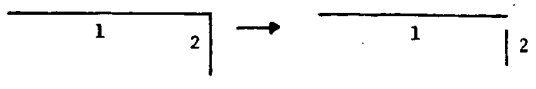
Case (1) Missing junction



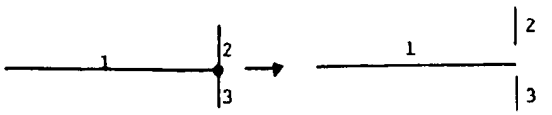
Case (2) Missing junction



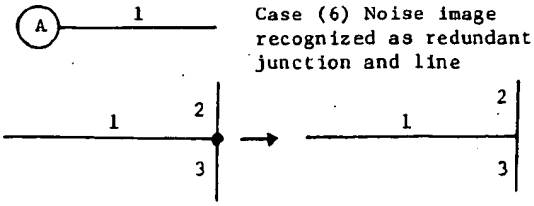
Case (3) Redundant junction



Case (4) Missing junction



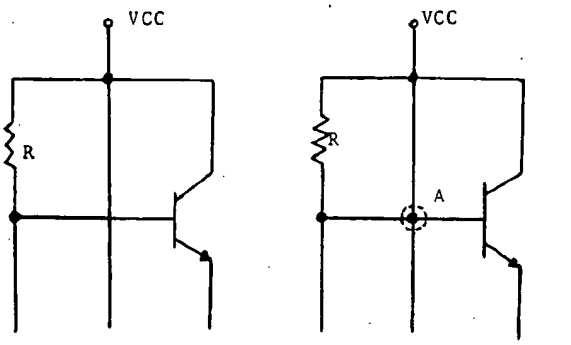
Case (5) Missing junction



Case (6) Noise image recognized as redundant junction and line

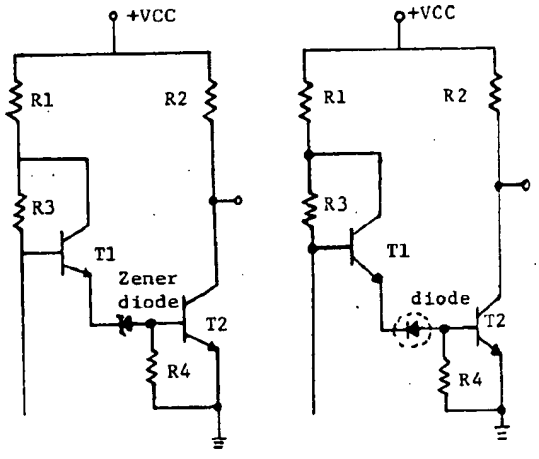
Case (7) Missing junction

Figure 4. Possible errors in interpreting junctions



(a) Legal Connection (b) Illegal Connection

Figure 5. Detecting errors by finding contradiction to circuit theories



(a) Legal circuit (b) Illegal circuit

Figure 6. Another example of detecting errors by finding contradiction to circuit theories

IV. CAPTURED CAD DATABASE VERIFICATION SYSTEM

Since the domain knowledge for recognizing circuit errors in CAD database is poorly-structured and case-oriented, knowledge-based expert system provides a good approach to the design of the verification system. The proposed knowledge-based system for database verification builds upon our previous work on pattern-directed, database-driven approaches to knowledge-based system design [5,8]. In this system we also make use of the concept of rule-based expert system design [9-11].

The architecture of the knowledge-based verification system is shown in Figure 7, which consists of six major subsystems: AUTORED generated CAD database, error identification scheme, context generator, working memory, knowledge base, and inference mechanism. The inference mechanism consists of inference rule selector, pattern matching, outcome evaluator, and database operators. When the type of error in the captured CAD database is identified, the influence rule selector fetches a subset of rules from the knowledge base which are associated with the error type. The identified error information is used by the context generator to select a subset of context generation rules for fetching erroneous data record as generated context to be temporarily stored in the working memory. The error type information also establishes a goal for the inference mechanism. The basic principle is iterative matching of generated contexts with appropriate patterns in the knowledge base. For example, if the error identifier detects that the type of a transistor is missing from the component data record, the system will fetch a set of rules in the knowledge base which are associated with the properties of transistors. If more than one rules are matched, the outcome evaluator will choose the one with highest score of similarity measure. The selected rule will activate a number of database operations specified in the action part of the rule. The activated data operations will change the contents of the working memory and/or the CAD database. The inference process terminates when the detected errors are corrected or all the rules have been selected.

Knowledge-base of the Verification System

Circuit diagram recognition and interpretation of AUTORED makes use of basic geometric features such as shape, length, terminals, and number of pixels to identify circuit elements. The interpretation result is dependent on the quality of the raw images of the circuit diagram. Any noise or distortion in the image resulted from light sources or camera problems is likely to cause interpretation errors. In order to verify these errors, image independent domain knowledge is provided for interpretation and inference. Domain knowledge in this system is defined as the knowledge required to verify the errors in the circuit based on circuit-level reasoning process. The reasoning process depends on either circuit theory or expert judgement to determine if a certain situation is going to happen. In our system, the domain knowledge is extracted from numerous design books, circuit designers' expertise, and accumulated experience from the AUTORED interpretation process. All of those circuit analysis tools are manually encoded as production rules and stored in the knowledge base. Following is the brief introduction of several sources for the knowledge base:

(1) Design conventions
Circuit designers usually follow design conventions for representing and identifying the circuit elements. For example, input terminals are usually placed at the left-hand side of the diagram while power source is located at the top or bottom of the circuit. Certain components are often used together in the same design, while others are not allowed to use at the same time. For instance, JFETs are usually combined with bipolar transistors in high-frequency circuit design. On the other hand, MOS transistors are seldom used with bipolar transistors in the same design. These design conventions are accumulated through consultation with circuit designers.

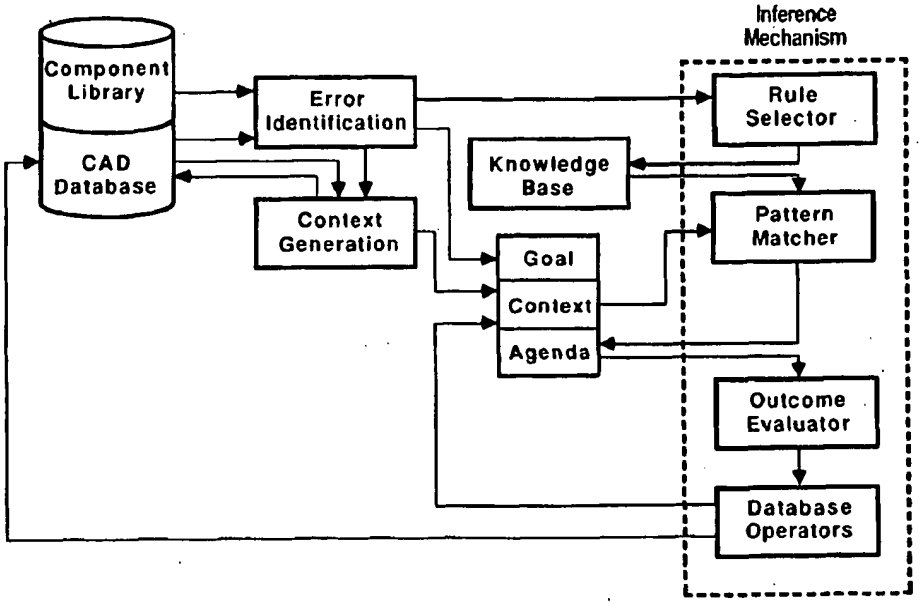


Figure 7. Function Flow of the Knowledge-based Verification System

(2) Electrical properties of the circuit
Analyzing electrical properties is another way to identify circuit elements under ambiguous situations. Properties such as direction of current flow, voltage drop, and on/off switching conditions can be effectively used in identifying circuit elements or detecting an error. The type of a bipolar transistor, for example, can be determined by analyzing the direction of current flow and voltage bias of the circuit.

(3) Accumulated experience of AUTORED
The intelligence level of AUTORED may be gradually augmented by accumulating experience in eliminating repeated errors. Corrected failures in the interpretation process of AUTORED may be recorded and encoded as inference rules for storage in the knowledge base. When a circuit is divided into several pages which are read separately by AUTORED, line segments connected outward to other pages will be interpreted as redundant lines and may be removed from the data file. Thus, AUTORED is unable to handle page connections. However, if we introduce the inference rule,

IF line A and line B have same y coordinates, and they are located at the opposite sides of two consecutive circuit pages,
THEN merge line A and B and update the data records of the connected elements.

AUTORED is enabled to handle interconnected pages of the whole circuit diagram.

Automated database-driven context generation

In conventional expert systems, contexts are generated through a user-driven question-answering process. The system utilizes the contexts to match the production rules to determine which actions should be taken under the current circumstances. In our knowledge-based verification system, the contexts are a set of descriptions or symptoms of a mis-recognized circuit element which needs to be identified. The contexts are automatically derived from the CAD database by a context generation process.

The database-driven context generation process consists of a set of context generation "unit operation". Each unit operation is designed for retrieving a certain feature (context) of the misinterpreted circuit element. Three kinds of data operations are utilized in each context generation unit: data fetching, data matching, and context generation. Data fetching is the process to fetch data from the CAD database. It includes direct data fetching and indirect data fetching. Direct data fetching fetches the data from the specified CAD data record. The format for direct data fetching operation is represented as:

$$DFOP_i(E_j) \Longrightarrow D_{ij}$$

where $DFOP_i$ represents the ith direct data fetching operator; E_j denotes the data record location of the jth circuit element; D_{ij} denotes the data (features) of the jth circuit element retrieved from the CAD database by the ith fetching operator.

Indirect data fetching fetches the data record location for direct data fetching. The format of indirect data fetching is:

$$IDFOP_i(E_j) \Longrightarrow E_{ij}$$

where $IDFOP_i$ denotes the ith indirect data fetching operator; E_j denotes the data record location of the jth circuit element; E_{ij} denotes a retrieved data record for further data operations.

The second data operation in context generation is called a data matching operation. The format is:

$$MATCH(DFOP_i(E_j),P_i)$$

or

$$MATCH(IDFOP_i(E_j),P_i)$$

where P_i denotes a pattern which is to be matched with data D_{ij} retrieved by direct fetching or with contents of E_{ij} retrieved by indirect fetching.

The last data operation for context generation is called context generation process. It is simply a data operation to add a new element (attribute-value pair) to the working memory if every condition is matched. Following is a format for a complete context generation unit:

IF $MATCH(OP_1(E_1),P_1)$ is true &
$MATCH(OP_2(E_2),P_2)$ is true &
•
•
$MATCH(OP_n(E_n),P_n)$ is true

THEN $ADD(A_i,V_i)$

A context generation rule is shown below for illustration:

MATCH(CATEGORY(CONNECT(E,n)),GROUND)
-----> ADD(tn,G)

This rule is employed to identify what kind of components are connected to a specified element E. CONNECT(E,n) is an indirect data fetching operation which fetches the data record location of the component which is connected to the nth terminal of the component. CATEGORY is a direct data fetching element which fetches the data entry concerning the category of the component. If the fetched data matches with the pattern "GROUND", it means that the nth terminal of E is connected to ground. The attribute-value pair (tn,G) is then added in the working memory. Some of frequently used context generation rules and database operators are listed below:

Context Generation Rules

MATCH(CATEGORY(E),BIPOLAR) ---> ADD(category,B)
MATCH(CNUMBER(E),N) ---> ADD(con,N)
MATCH(ACOORDINATE(E),X) ---> ADD(xcoor,X)
MATCH(CATEGORY(CONNECT(E,n)),GROUND)
---> ADD(tn,G)
MATCH(CATEGORY(E),BJT) &
MATCH(ORIENTATION(E),EAST) ---> ADD(term1,BASE)

Data Fetching Operators for Context Generation

a)Direct data fetching

CATEGORY(E)	Fetch the category data of component E.
CNUMBER(E)	Aggregate the total number of circuit components connected to E.
ORIENTATION(E)	Fetch the orientation of component E.
XCOORDINATE(E)	Fetch the x coordinate of component E.

b)Indirect data fetching

CONNECT(E,n)	Fetch the data allocation of all the circuit components connected to the nth terminal of component E.

Pattern-directed inference process

After contexts are generated and stored in working memory, the system starts the inference process to determine what actions are to be taken. The knowledge base contains inference rules of circuit verification which are partitioned into numerous subsets. Each subset of inference rules is responsible for resolving a certain type of circuit interpretation ambiguity. An inference rule is represented as:

$$\begin{array}{ll} \text{IF} & C_1 \& C_2 \cdots \& C_n \\ \text{THEN} & A_1 \& A_2 \cdots \& A_m \end{array}$$

The left side of the inference rule contains several rule patterns which are represented as attribute-value pairs. If all the attribute-value pairs at the left side match with the corresponding attribute-value pairs stored in working memory, all the actions at the right of the rule are activated. The right side of the inference rule contains a set of actions which are in the form of data operations. The data operations are used to generate new contexts or to modify the contents of CAD database and are represented by the format:

$$OP_i\,(p_1, p_2, \cdots p_2, \cdots p_n)$$

where OP_i denotes the database operator, p_i denotes the ith parameter of the database which can be either an access key or a data value. Samples of database operators used in inference are:

REPLWM((a,b),(a,c))	replace attribute-value pair (a,b) in the working memory with pair (a,c).
UPDATE(a,b,c,ov,nv)	update the value of data record in CAD database. (a is file No., b is data No., c is data entry No., Ov is original data contents, nv is new data contents).
REPLDB(a,b,ov,nv)	replace the original data in data record with new value.
DELETE(a,b)	delete data record b in file a.
ENTER(a,b,v)	enter new data record b (with data contents v) in file a.

During the inference process, it is very likely that more than one rule are matched. The inference mechanism uses outcome evaluator to resolve conflicts. All the matched rules are ordered and selected on the basis of a similarity measure:

$$S = \frac{number\ of\ matched\ patterns}{total\ number\ of\ rule\ patterns}$$

The similarity measure ranges from 1 (all the rule patterns are matched) to 0 (none of the rule patterns are matched). The rule with the highest similarity is selected and activated. To illustrate the inference process, we consider an example of identifying the type of a bipolar transistor when the arrowhead of the transistor symbol is missing, as shown in Figure 8. With missing arrowhead, AUTORED ia unable to identify the type of transistor. By detyecting that the data entry of transistor type is missing in the database, the system will make use of circuit theory to identify the circuit type. Since the type of bipolar transistor can be either NPN or PNP, the goal of the inference process is set, as (type,N) or (type,P). After the goal is set, the context generator starts to generate information regarding the transistor. The generated contexts are stored in the working memory. The rule selector decides which subset of the inference rules will be retrieved. Since the error detector indicates that the transistor type is missing and the order of transistor terminals is unknown, the rule sector decides to access rule set 1 (a rule set designed to identify the terminals of bipolar transistor) first. After the pattern matching process, rule set 1-1 is patterned and triggered. After the rule is triggered, the information concerning

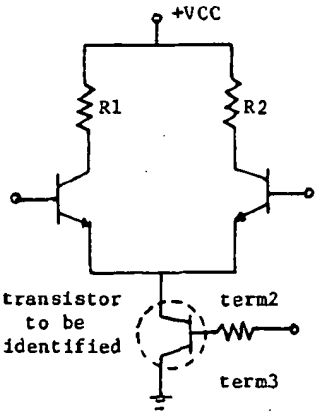


Figure 8. An example of circuit component identification

the order of the transistor terminals are added to working memory. With the new information, a different rule set will be retrieved. The inference continues until the preset goal is matched or all the rule sets have been triggered. A complete list of the inference procedure for the given example is given below:

Goal: (type,N) or (type,P)

Contents of working memory (before rule inference):

(type,NULL), (term1,BASE), (term2,NULL), (term3,NULL)
(category,B), (t1,R), (t2,BT3E), (t2i,PV), (t3,G)

Selector — Select rule set 1 (a rule subset designed to identify the terminals of bipolar transistors).

Matched rules:

Rule 1-1 --- (category,B) & (t3,G) —>
REPLWM((term2,$),(term2,COLLECTOR))
REPLWM((term3,$),(term3,EMITTER))

Rule 1-2 --- (category,B) & (t2,BT3E) —>
REPLWM((term2,$),(term2,COLLECTOR))
REPLWM((term3,$),(term3,EMITTER))

Evaluation --- Select rule 1-1
Triggered Actions:
REPLWM((term2,$),(term2,COLLECTOR))
REPLWM((term3,$),(term3,EMITTER))

----- Iteration 2 -----

Contents of Working Memory (after first iteration):

(type,NULL), (term1,BASE), (term2, COLLECTOR),
(term3,EMITTER),(category,B), (t1,R), (t2, BT3E),
(t2i,PV),(t3,G)

Selector — Select rule subset 2

Matched rule:

Rule 2-1 (category,B) & (term2, COLLECTOR) & (t2i,PV)
—> REPLWM ((type,$), (type,N))
UPDATE (1,ID,7,$,N)

Evaluation — Select rule 2-1

Triggered Actions: REPLWM ((type,$), (type,N))
UPDATE (1,ID,7,$,N)

----- Iteration 3 -----

Contents of working memory (after iteration 2);

(type,N), (term1,BASE), (term2, COLECTOR), (term3, EMITTER), (category,B), (t1,R), (t2,BT3E), (t2i,PV), (t3,G)

Triggered actions: process is terminated; the value of transistor type is determined and entered into the component data record.

V. EXPERIMENTAL RESULTS

A knowledge-based system for verifying the CAD database generated by an automatic schematic capture system has been implemented on a VAX-11/750 minicomputer under VMS operation system. Some experimental results are presented in this section. The raw image of an electronic circuit diagram is shown in Figure 9. AUTORED interprets the circuits and generates the component list as shown in Figure 10. Due to the ambigyuity in raw image, as many as 8 redundant lines and 34 redundant junctions are generated. Moreover, nine of the components are either partially recognized or misinterpreted. After the database verification process, all of the redundant records are removed and erroneous records are corrected. Figure 11 shows the computer generated statistics of the verification process. The generated table provides noot only the total number of corrected errors but also the individual ID number of the data record corrected. Figure 12 illustrates the correct component list and netlist extracted from the verified data files. The verification results are summarized as follows:

Connection lines - 8 redundant lines removed.

Junctions - 34 redundant junctions removed.

Components - 5 redundant components removed:

(ID No. 10,11,15,17,18)

Misinterpreted components are corrected as:

ID	Original Category	Updated Category	Added Information
1	port	power	
4			type N
5			type N
6	port	output	
7			type N
8			type N
12	port	input	
13	port	output	
19	port	power	

VI. CONCLUSION

In this paper, we present the design of a knowledge-based expert system for the verification of CAD database generated by an automatic schematic capture system-AUTORED. Database-driven pattern-directed inference technique is employed to identify erroneous data records due to misrecognition. This knowledge-based expert system has been implemented on a VAX-11/750 minicomputer equipped with a video camera to read circuit diagrams into the computer. Integration of this knowledge-based expert system into the AUTORED system greatly improves the accuracy of the automatically generated CAD database. It is hoped that the fundamental principles presented in this paper will form the foundation for the design of intelligent CAD systems for design automation.

REFERENCES

[1] H. M. Bayegan and E. Aas, "An Integrated System for Interactive Editing of Schematics, Logic Simulation and PCB Layout Design," Proc. 14th Design Automation Conference, 1977, pp. 1-6.

[2] G. Odawara, S. Kurishima, H. Aoyama and Y. Kanaya, "PAS-CIP: An Interactive Logic Design System," Proc. 18th Design Automation Conference, 1981, pp. 443-449.

[3] J. T. Tou and J. M. Cheng, "AUTORED: An Automated Electronic Diagram Reading Machine," Proc. of 1983 IEEE Int. Conf. Computer-aided Design, 1983.

[4] J. T. Tou, C. L. Huang, and W. H. Li, "Design of a Knowledge-based System for Understanding Electronic Circuit Diagrams," Proc. of First conference on Artificial Intelligence Applications, 1984.

[5] J. T. Tou, "MEDIKS - A Medical Knowledge System," Proc. of the 31st Annual Conference on Engineering in Medicine and Biology, 1978.

[6] J. T. Tou, "Knowledge Engineering," *International Journal of Computer and Information Sciences*, Vol. 9, No. 4, 1980.

[7] J. T. Tou, "Application of Pattern Recognition to Knowledge System Design and Diagnostic Inference," in *Pattern Recognition Theory and Applications*, edited by J. Rittler, K. S. Fu and L. F. Pan, D. Reidel Publishing Co. 1982.

[8] J. T. Tou and J. M. Cheng, "Design of a Computer-based Expert System for Applications in Agriculture," Proc. of the IEEE Symposium on Automating Intelligent Behavior, 1983.

[9] D. A. Waterman & F. Hayes-Roth (Ed.), *Pattern-Directed Inference Systems*, Academic Press Inc., New York, 1978.

[10] J. McDermott, "Domain Knowledge and the Design Process," Proc. 18th Design Automation Process, 1981, pp. 580-588.

[11] T. J. Kowalski and D. E. Thomas, "The VLSI Design Automation Assistant: Prototype System," Proc. 20th Design Automation Conference, 1983, pp. 479-483.

[12] J. Markus, *Sourcebook of Electronic Circuits*, McGraw-Hill Book Co., 1968.

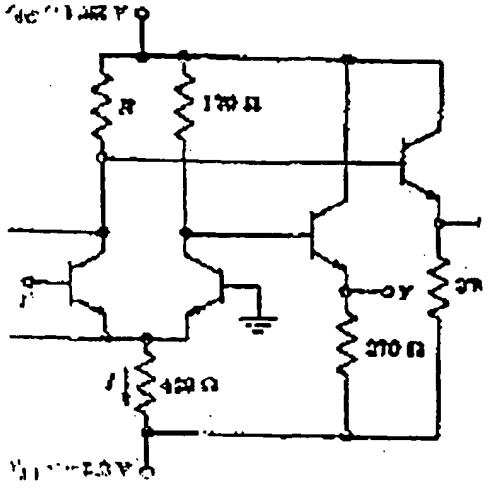


Figure 9. Input image of an example circuit

COMPONENT LIST
BEFORE CIRCUIT VERIFICATION

ID	NAME	ORIENT	CATEGORY	TYPE
1	S1	E	PORT	
2	E1	N	RESISTOR	
3	E2	N	RESISTOR	
4	B1	E	BIPOLAR	
5	B2	E	BIPOLAR	
6	S2	E	PORT	
7	B3	E	BIPOLAR	
8	B4	W	BIPOLAR	
9	E3	N	RESISTOR	
10	S3	E	PORT	
11	Y1	E	DIODE	
12	S4	E	PORT	
13	S5	E	PORT	
14	E4	N	RESISTOR	
15	S6	E	PORT	
16	E5	N	RESISTOR	
17	S7	E	PORT	
18	E6	N	RESISTOR	
19	S8	E	PORT	
20	G1	E	GROUND	

Figure 10. Component list generated by AUTORED before database verification

```
* OVERALL STATISTICS OF THE CIRCUIT *
* BEFORE CIRCUIT VERIFICATION *

TOTAL NUMBER OF JUNCTIONS -- 52
TOTAL NUMBER OF CONNECTION LINES -- 32
TOTAL NUMBER OF COMPONENTS -- 20

* DIAGNOSIS OF THE CIRCUIT DIAGRAM *

8 REDUNDANT LINES WERE REMOVED:
14 18 16 11 12 4 6 13

34 REDUNDANT JUNCTIONS WERE REMOVED:
1 2 3 4 11 15 18 20 21 22 23 24 25
27 29 30 35 36 37 38 39 40 41 43 45
47 50 51 6 19 28 33 34 42

5 REDUNDANT COMPONENTS WERE REMOVED:
10 11 15 17 18

* OVERALL STATISTICS OF THE CIRCUIT *
* AFTER CIRCUIT VERIFICATION *

TOTAL NUMBER OF JUNCTIONS -- 18
TOTAL NUMBER OF CONNECTION LINES -- 24
TOTAL NUMBER OF COMPONENTS -- 15

DIGITAL COMPONENTS -- 0

ANALOG COMPONENTS (ACTIVE ELEMENTS) --
    4 BIPOLAR TRANSISTORS

ANALOG COMPONENTS (PASSIVE ELEMENTS) --
    5 RESISTORS

POWER AND I/O ELEMENTS --
    2 POWER SOURCES
    1 GROUND ELEMENT
    1 INPUT
    2 OUTPUTS
```

Figure 11. CAD database verification statistics generated by the knowledge-based circuit verification system

COMPONENT LIST
AFTER CIRCUIT VERIFICATION

ID	NAME	ORIENT	CATEGORY	TYPE
1	P1	E	POWER	
2	E1	N	RESISTOR	
3	E2	N	RESISTOR	
4	B1	E	BIPOLAR	N
5	B2	E	BIPOLAR	N
6	T1	E	OUTPUT	
7	B3	E	BIPOLAR	N
8	B4	W	BIPOLAR	N
9	E3	N	RESISTOR	
10	U1	E	NOISE	
11	U2	E	NOISE	
12	I1	E	INPUT	
13	T2	E	OUTPUT	
14	E4	N	RESISTOR	
15	U3	E	NOISE	
16	E5	N	RESISTOR	
17	U4	E	NOISE	
18	U5	N	NOISE	
19	P2	E	POWER	
20	G1	E	GROUND	

NET LIST AFTER CIRCUIT VERIFICATION

NET 1	NET 2	NET 3	NET 4
B1-1	E1-1	T1-1	B2-1
E2-1	B2-3	B1-2	E1-2
B3-3	P1-1	E3-2	B4-3
	E2-2		
	B1-3		

NET 5	NET 6	NET 7	NET 8
T2-1	E5-1	E5-2	B4-1
B2-2	B3-2	P2-1	G1-1
E4-2	E5-3	E4-1	
	B4-2	E3-1	

Figure 12. Component list and net list generated after database verification